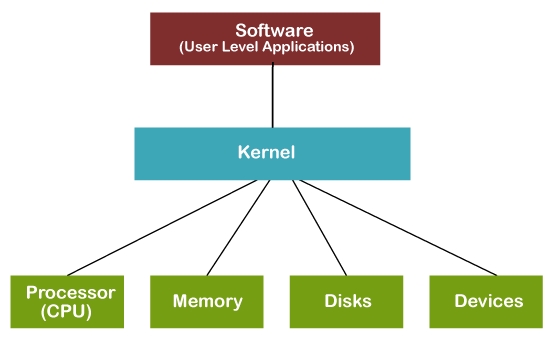
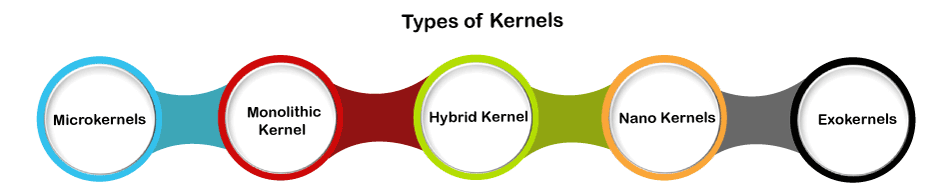
**What is Kernel in Operating System?**

A kernel is an important part of an OS that manages system resources. It also acts as a bridge between the software and hardware of the computer. Kernel is also responsible for offering secure access to the machine’s hardware for various programs.



* Kernel is the core part of an OS(Operating system); hence it has full control over everything in the system. Each operation of hardware and software is managed and administrated by the kernel.
* It acts as a bridge between applications and data processing done at the hardware level. It is the central component of an OS.



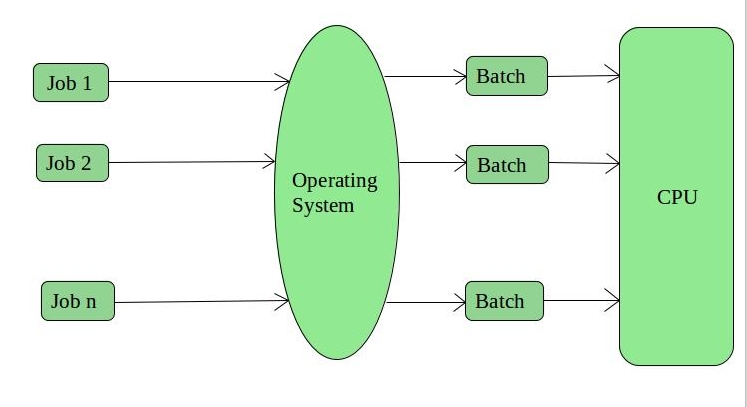
## What is Microkernel?

**Microkernel** is a software or code which contains the required minimum amount of functions, data, and features to implement an operating system

A microkernel comprises only the core functionalities of the system. A component is included in the Microkernel only if putting it outside would interrupt the functionality of the system

# Types of Operating Systems

**1. Batch Operating System –**   
This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.



**Advantages of Batch Operating System:**

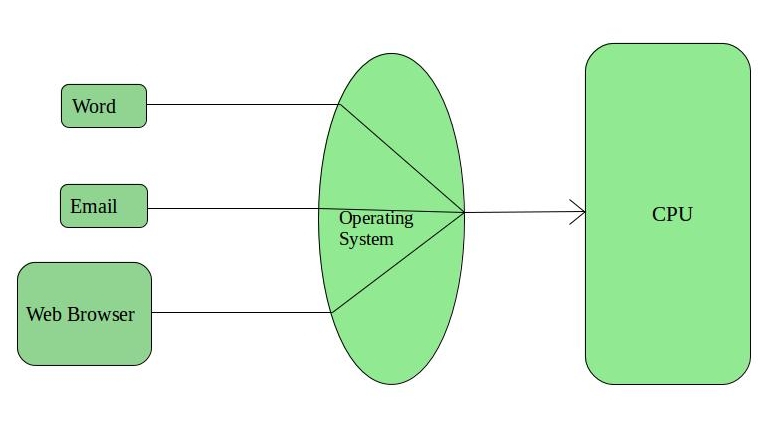
* It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
* Multiple users can share the batch systems
* The idle time for the batch system is very less
* It is easy to manage large work repeatedly in batch systems

**Disadvantages of Batch Operating System:**

* The computer operators should be well known with batch systems
* Batch systems are hard to debug
* It is sometimes costly
* The other jobs will have to wait for an unknown time if any job fails

**Examples of Batch based Operating System:** Payroll System, Bank Statements, etc.

**2. Time-Sharing Operating Systems –**   
Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



**Advantages of Time-Sharing OS:**

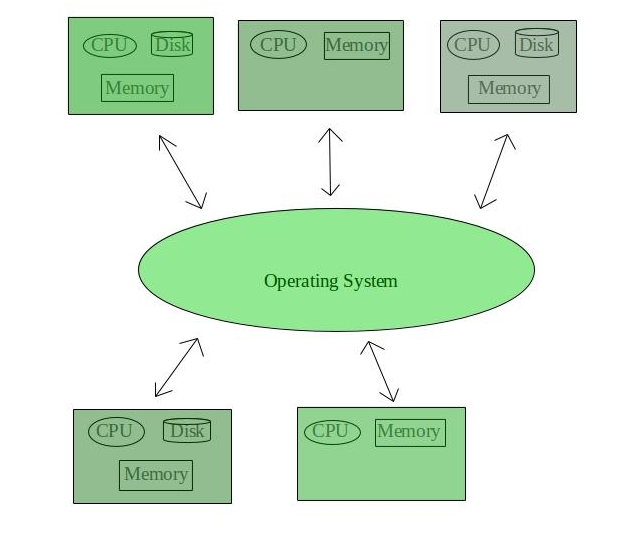
* Each task gets an equal opportunity
* Fewer chances of duplication of software
* CPU idle time can be reduced

**Disadvantages of Time-Sharing OS:**

* Reliability problem
* One must have to take care of the security and integrity of user programs and data
* Data communication problem

**Examples of Time-Sharing OSs are:** Multics, Unix, etc.

**3. Distributed Operating System –**   
These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as **loosely coupled systems** or distributed systems. These system’s processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network. 



**Advantages of Distributed Operating System:**

* Failure of one will not affect the other network communication, as all systems are independent from each other
* Electronic mail increases the data exchange speed
* Since resources are being shared, computation is highly fast and durable
* Load on host computer reduces
* These systems are easily scalable as many systems can be easily added to the network
* Delay in data processing reduces

**Disadvantages of Distributed Operating System:**

* Failure of the main network will stop the entire communication
* To establish distributed systems the language which is used are not well defined yet
* These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

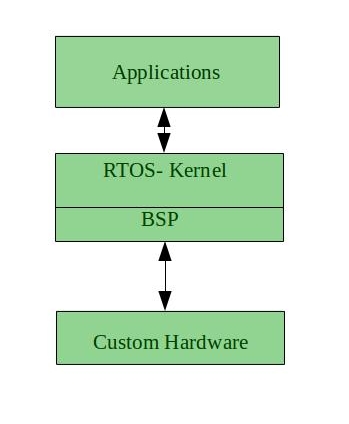
**Examples of Distributed Operating System are-** LOCUS, etc.

**4. Real-Time Operating System –**   
These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.

**Real-time systems** are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

**Two types of Real-Time Operating System which are as follows:**

* **Hard Real-Time Systems:**   
  These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident. Virtual memory is rarely found in these systems.
* **Soft Real-Time Systems:**   
  These OSs are for applications where for time-constraint is less strict.



**Advantages of RTOS:**

* **Maximum Consumption:** Maximum utilization of devices and system, thus more output from all the resources
* **Task Shifting:** The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
* **Focus on Application:** Focus on running applications and less importance to applications which are in the queue.
* Real-time**operating system in**the **embedded system:** Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
* **Error Free:** These types of systems are error-free.
* **Memory Allocation:** Memory allocation is best managed in these types of systems.

**Disadvantages of RTOS:**

* **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
* **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
* **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
* **Device driver and interrupt signals:** It needs specific device drivers and interrupts signals to respond earliest to interrupts.
* **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

**Examples of Real-Time Operating Systems are:** Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

**Types of OS Installation**

Installation refers to the particular configuration of a software or hardware with a view to making it usable with the computer. A soft or digital copy of the piece of software (program) is needed to install it.

* Making sure that necessary [system requirements](https://en.wikipedia.org/wiki/System_requirements) are met
* Checking for existing versions of the software
* Creating or updating program [files](https://en.wikipedia.org/wiki/Computer_file) and folders
* Adding configuration data such as [configuration files](https://en.wikipedia.org/wiki/Configuration_file), [Windows registry](https://en.wikipedia.org/wiki/Windows_registry) entries or [environment variables](https://en.wikipedia.org/wiki/Environment_variable)
* Making the software accessible to the user, for instance by creating [links, shortcuts](https://en.wikipedia.org/wiki/Computer_shortcut) or [bookmarks](https://en.wikipedia.org/wiki/Bookmark_(world_wide_web))
* Configuring components that run automatically, such as [daemons](https://en.wikipedia.org/wiki/Daemon_(computing)) or [Windows services](https://en.wikipedia.org/wiki/Windows_service)
* Performing [product activation](https://en.wikipedia.org/wiki/Product_activation)
* Updating the software versions

**Types**

**Attended installation**

On [Windows](https://en.wikipedia.org/wiki/Windows) systems, this is the most common form of installation. An installation process usually needs a user who attends it to make choices, such as accepting or declining an [end-user license agreement](https://en.wikipedia.org/wiki/End-user_license_agreement) (EULA), specifying preferences such as the installation location, supplying passwords or assisting in [product activation](https://en.wikipedia.org/wiki/Product_activation).

**Silent installation**

Installation that does not display messages or windows during its progress. "Silent installation" is not the same as "unattended installation" (see below): All silent installations are unattended but not all unattended installations are silent.

**Unattended installation**

Installation that is performed without user interaction during its progress or with no user present at all. One of the reasons to use this approach is to automate the installation of a large number of systems. An unattended installation either does not require the user to supply anything or has received all necessary input prior to the start of installation

**Headless installation**

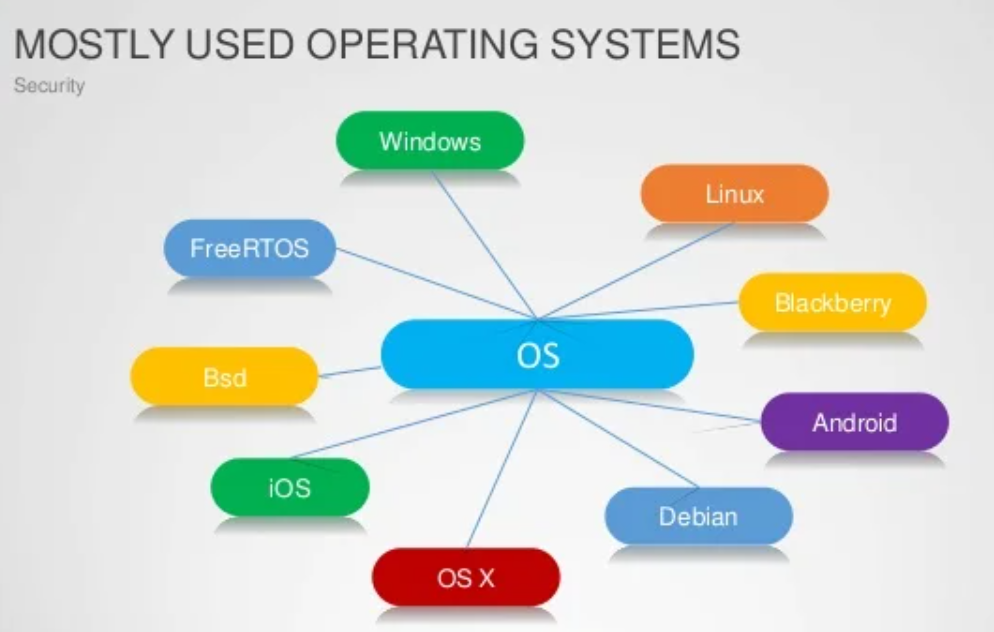
Installation performed without using a [computer monitor](https://en.wikipedia.org/wiki/Computer_monitor) connected. In attended forms of headless installation, another machine connects to the target machine (for instance, via a [local area network](https://en.wikipedia.org/wiki/Local_area_network)) and takes over the display output

**Scheduled or automated installation**

An installation process that runs on a preset time or when a predefined condition transpires, as opposed to an installation process that starts explicitly on a user's command

**Network installation**

Network installation, shortened netinstall, is an installation of a program from a [shared network resource](https://en.wikipedia.org/wiki/Shared_resource)



## What is Mobile Operating System?

A **mobile OS** allows application software to operate on mobile devices. It is similar to desktop OS in certain ways, but it is simpler and lighter in comparison.

Smartphone operating systems include

* Windows Mobile,
* Palm webOS,
* Symbian OS,
* RIM's Blackberry,
* Android,
* iPhone OS,
* Maemo.

Features of the Mobile Operating System

There are various features of a Mobile Operating System. Some features of the Mobile Operating System are as follows:

1. It is very easy to understand and utilize. The graphics of the mobile operating system is very attractive
2. It provides better apps to use.

## What is Desktop Operating System?

The desktop OS is the environment in which a user manages a personal computer.

It helps in the management of system hardware and software resources.

For example,

* Windows,
* Mac OS, and
* various Linux distributions.

It supports basic features, including task scheduling, printing, I/O, peripheral control, and memory allocation.

**Advantages**

1. The operating system acts as a bridge between the user and the system hardware. It enables users to enter data, process it, and view the results.
2. It enables data and relevant information to be shared with other users through Players, Printers, Modems, and Fax Machines.

|  |  |  |
| --- | --- | --- |
| **Features** | **Mobile Operating System** | **Desktop Operating System** |

|  |  |  |
| --- | --- | --- |
| Definition | It is a type of operating system that allows application software to operate on mobile devices. | It is the environment in which a user handles a personal computer. |
| Memory Requirement | It needs minimum RAM to optimize. | It needs huge memory to operate. |
| Storage | It uses a flash drive to store the data. | It uses hard drives and flash drives to store data. |
| Boot Time | It takes less time to boot. | It takes much time to boot. |
| Purpose | It handles cellular and wireless connectivity and device access. | It handles the software and hardware resources of the system. |
| Power | It is optimized to work under minimal power needs and has a feature to prevent energy loss. | It is not readily optimized for energy loss. |
| Interface | It runs on touchscreen or touchpad devices. | It runs through many input devices, including a mouse, keyboard, etc. |
| Example | Some examples of the Mobile OS are Apple iOS, Google Android, Bada, Palm OS, Symbian OS, Windows Mobile OS, Blackberry OS, iPhone, Harmony OS, WebOS, etc. | Some examples of the desktop OS are Windows 10, MacOS, Windows Vista, etc. |

**User Interface**

A **User interface (UI)** facilitates communication between an application and its user by acting as an intermediary between them. Each application including the operating system is provided with a specific UI for effective communication. The two basic function of a user interface of an application is to take the inputs from the user and to provide the output to the users. However, the types of inputs taken by the UI and the types of output provided by the UI may vary from one application to another.

A user interface of any operating system can be classified into one of the following types:

1. Graphical user interface (GUI)
2. Command line user interface (CLI)

**Graphical user interface (GUI)**

The graphical icon provided in the UI can be manipulated by the users using a suitable pointing device such as a

* mouse,
* trackball,
* touch screen
* light pen.
* keyboard can also be used to manipulate these graphical icons.

 interact with the operating system by issuing some specific commands.

**Command line user interface (CLI)**

* Enables the users to interact with the operating system by issuing some specific commands.
* user has to memorize so many commands or to refer book for different commands

C:\>DATE

The current date is: Sun 03/04/2022

### FUNCTIONS OF OPERATING SYSTEM

### Security

The operating system uses a password protection to protect user data it also prevents unauthorized access to programs and user data, but for external functionality we need to install malware software to protect the system.

### Control over system performance

The operating system monitors overall system setup to help in improving the performance and it also records the response time between service requests and system response so that it has a complete view of the system. This can help improve performance by providing important information that is needed at the time of troubleshooting problems.

### Job Accounting

Operating systems always keep track of time and resources that are used by various tasks and users, this information can be used to track resource usage for a particular user or a group of users.

### Error detecting aids

Operating systems constantly monitor the system which helps us to detect errors and also avoid the malfunctioning of computer systems.

### Coordination between other software and users

Operating systems help in coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.

### Memory Management

The operating system controls the primary memory or main memory. Primary memory is a large array of bytes or words where each byte or word is assigned a certain address. It is a fast storage, and it can be accessed directly by the CPU which is present inside the system. If a program wants to be executed, it should be first loaded in the main memory.

The following activities are performed by operating system for memory management −

* It keeps track of primary memory.
* Memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used.
* In multiprogramming, the OS decides for how long the process must stay and the order in which processes are granted access to memory.
* It allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated.

### Processor Management

The OS manages the order in which processes have access to the processor, and how much processing time that each process must stay in the multiprogramming environment. This is called process scheduling.

The following activities are performed by operating system for processor management −

* Keeps track of the status of processes.
* The program to track the status is known as traffic controller.
* It allocates the CPU and deallocates the processor when it is not required.

### Device Management

An OS manages device communication through respective drivers.

The following activities are performed by the operating system for device management.

* Keeping track of all devices connected to the system.
* The OS designates a program that is responsible for every device which is called the Input/output controller.
* It decides which process gets access to which device and for how long. It then allocates the devices in an effective and efficient way and de-allocates devices when they are not required.

### File Management

A file system is arranged into directories for efficient navigation and usage. These directories contain other directories and other files.

The following activities are performed by operating system for file management activities −

* It keeps track of where information is stored, user access settings and status of every file and more.
* These facilities are called the file system.

User application programs can only interact with the system hardware with the help of an operating system. Operating system provides an environment to different types of application and programs in which they can perform their useful task , it itself doesnot provide any functionality.

Operations of operating system are :

1. process management,

2. memory management,

3. device management and

4. file management

Process Management

The Operating system manages the processes, it assigns the processor to process the task ata time,which is termed as process scheduling. To do this it uses different types of algorithms like

* FCFS (first come first served),
* SJF (shortest job first),
* priority scheduling,
* round robin scheduling etc.

To handle processes in process management many scheduling queues are used.

1. As the processes enter the system, they are queued under the job queues.
2. When the processes are ready to execute in the main memory, they are sent to ready queue.
3. Processes that needs I/O device are sent to device queue.

Memory Management

The role of Memory Management is very important in operating system. It allocates memory and switches the processes between disk and primary memory for execution whenever required.

Following activities are performed by operating system for memory management.

* it allocates memory to the processes by using some algorithms like best fit, first fit, and worst fit
* All memory allocations are being tracked by operating system, means what part of memory are in use and what parts are empty.
* Operating System also deallocate memory from processes when process gets terminated or if process does not require it.

Device Management

operating system handles many I/O devices such as mouse, keyboard, disk drive etc. To handle a specific device, operating system can be connected to different device drivers. Interface of the device and the device driver is device controller. All the I/O devices such as mouse, keyboard, disk drive etc. can be accessed by user applications by using device drivers.

File Management

To provide a uniform and systematic view of data storage, Files are used by the operating system.

All the files are mapped to non volatile physical devices, in order to protect data loss in case of system failure.

There are two ways to access Files

1. sequential access and

2. direct access

Sequential Access

In this Data in files are accessed or processed in serialised/sequential order.

Data is accessed one by one ie. one after the other.

some examples of file systems that uses sequemtial access are editors, compilers etc.

Direct Access

It is a Relative access ie. in this files are accessed in random order for read and write operations.

Modern operating systems are interrupt driven.

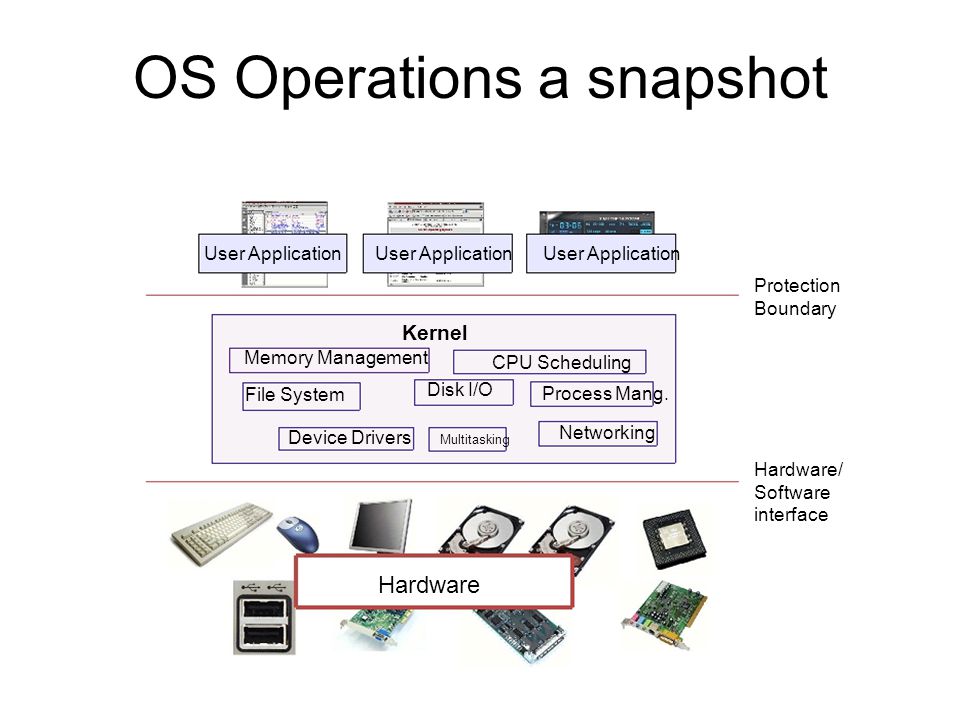
* If there are no processes to execute,
* no I/O devices to service, and no users to whom to respond,

an operating system will sit quietly, waiting for something to happen. Events are almost always signaled by the occurrence of an **interrupt** or a **trap**.

**Wondering!! what is trap ?**

A trap (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed.

* The interrupt-driven nature of an operating system defines that system's general structure. For each type of interrupt, separate segments of code in the operating system determine what action should be taken.
* An interrupt service routine is provided that is responsible for dealing with the interrupt. Since the operating system and the users share the hardware and software resources of the computer system, we need to make sure that an error in a user program could cause problems only for the one program that was running. With sharing, many processes could be adversely affected by a bug in one program. For example, if a process gets stuck in an infinite loop, this loop could prevent the correct operation of many other processes.
* More subtle errors can occur in a multiprogramming system, where one erroneous program might modify another program, the data of another program, or even the operating system itself. Without protection against these sorts of errors, either the computer must execute only one process at a time or all output must be suspect. A properly designed operating system must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.



**Dual-Mode Operation**

In order to ensure the proper execution of the operating system, we must be able to distinguish between the execution of operating-system code and user defined code. The approach taken by most computer systems is to provide hardware support that allows us to differentiate among various modes of execution. At the very least, we need two separate modes of operation: **user mode** and **kernel mode** (also called **supervisor mode**, **system mode**, or **privileged mode**). A bit, called the **mode bit**, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). With the mode bit, we are able to distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user.

* When the computer system is executing on behalf of a user application, **the system is in user mode**.
* However, when a user application requests a service from the operating system (via a system call), it must **transition from user to kernel mode** to fulfill the request. As we shall see, this architectural enhancement is useful for many other aspects of system operation as well.
* At system boot time, the hardware starts in **kernel mode.**
* The operating system is then loaded and starts user applications in **user mode**.
* Whenever a trap or interrupt occurs, the hardware switches from **user mode to kernel mode** (that is, changes the state of the mode bit to 0).
* Thus, whenever the operating system gains control of the computer, it is in **kernel mode**.
* The system always **switches to user mode** (by setting the mode bit to 1) before passing control to a user program.

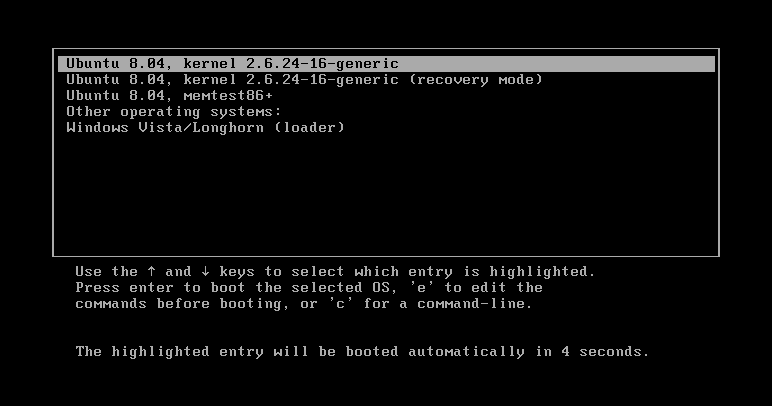
Functions of Operating System

Given below are the various functions of an Operating System:

* It helps with memory management. It keeps a track of the files being saved in the Main memory and the primary memory of the computer device
* Whenever a computer is turned on, the Operating system automatically starts to work. Thus, the booting and rebooting process of a computer device is also an important function of the OS
* It provides a user interface
* Managing of basic peripheral devices is done by the operating system
* Using the password protection option of an operating system, the data in the device can be kept secure
* It coordinates with the software and the user
* Easy navigation and organisation of files and programs are managed by the OS
* Any kind of program which needs to be run through the system is done by the operating system
* If any kind of error or bug is found during the program is detected using the operating system

## Booting

Booting is basically the process of starting the computer. When the CPU is first switched on it has nothing inside the Memory. In order to start the Computer, load the Operating System into the Main Memory and then Computer is ready to take commands from the User.



There are two types of booting:

### Cold Booting

A cold boot is also called **a hard boot.**It is the process when we first start the computer. In other words, when the computer is started from its initial state by pressing the power button it is called cold boot. The instructions are read from the ROM and the operating system is loaded in the main memory.

### Warm Booting

Warm Boot is also called **soft boot**. It refers to when we restart the computer. Here, the computer does not start from the initial state. When the system gets stuck sometimes it is required to restart it while it is ON. Therefore, in this condition the warm boot takes place. Restart button or CTRL+ALT+DELETE keys are used for warm boot.